DATALYXT

AUSGANGSLAGE

Wenn alles in einem Aufruf passiert

Eine ganze Aufgabe an ein Sprachmodell zu geben, ist schnell gebaut. Prüfen lässt sich daran wenig.

Wer eine ganze Aufgabe in einen Aufruf gibt, hat genau eine Stellschraube: den Prompt. Alles Übrige entscheidet das Modell unsichtbar.

Was gemeint ist

Ein Aufruf, eine Anweisung, ein Absatz Antwort. Das sieht sparsam aus und ist in einem Tag gebaut.

Beispiel: „Prüfe diese Rechnung gegen die Bestellung und sag mir, ob sie freigegeben werden kann."

Im Hintergrund steckt darin: auslesen, Positionen zuordnen, Differenz rechnen, Vertragsregeln anwenden, bewerten, formulieren. Sechs Teilaufgaben in einem Schritt, ohne Zwischenergebnis.

Fünf Folgen

1. **Ein Fehler lässt sich nicht verorten.**
   War es die Erkennung, die Zuordnung, die Rechnung oder die Bewertung? Sichtbar ist nur das Ergebnis.

2. **Es gibt nichts zu prüfen außer der Endaussage.**
   Ohne Zwischenergebnisse fehlt jede Stelle, an der ein Schema oder eine Regel greifen könnte.

3. **Jede Korrektur ist eine Textänderung.**
   Der Prompt wirkt auf alle Fälle zugleich. Was einen Fall verbessert, kann zehn andere verschlechtern.

4. **Es wird gerechnet, wo nicht gerechnet werden sollte.**
   Summen, Fristen und Schwellen entstehen im Sprachmodell: plausibel formuliert und ungeprüft.

5. **Der Aufwand ist nicht planbar.**
   Länge und Kosten der Antwort hängen am Eingabetext. Einen Punkt zum Abbrechen gibt es nicht.

Das Modell ist dabei nicht das Problem. Das Problem ist, dass niemand sieht, was es getan hat.

DATALYXT

ALTERNATIVE

Derselbe Auftrag, in Schritte zerlegt

Jeder Schritt hat eine Ein- und eine Ausgabe. Genau dort entsteht Kontrolle.

1 Beleg auslesen

Das Sprachmodell liest Nummer, Datum, Positionen und Beträge aus dem Dokument.
Die Ausgabe wird gegen ein Schema geprüft: erwartete Felder, Datentypen, Wertebereiche.
Kontrolle: Was nicht ins Schema passt, geht in den Ausnahmekorb statt weiter.

2 Positionen zuordnen

Das Programm ordnet Rechnungs- und Bestellpositionen einander zu.
Die Zuordnungsregel steht im Code und gilt für jeden Vorgang gleich.
Kontrolle: Jede Zuordnung ist im Protokoll nachvollziehbar.

3 Differenz berechnen

Das Programm rechnet. Das Sprachmodell ist an dieser Stelle nicht beteiligt.
Gleiche Eingaben ergeben jedes Mal dasselbe Ergebnis.
Kontrolle: Der Rechenweg lässt sich nachvollziehen und im Zweifel von Hand prüfen.

4 Abweichung einordnen

Ein Regelwerk entscheidet anhand von Schwellenwerten, ob die Abweichung zulässig ist.
Das Sprachmodell erklärt den Befund in Worten, es trifft ihn nicht.
Kontrolle: Die Schwellenwerte sind sichtbar und änderbar, ohne den Prompt anzufassen.

5 Ergebnis formulieren und freigeben

Das Sprachmodell schreibt den Vermerk aus den geprüften Werten.
Eine benannte Person gibt frei, bevor etwas wirksam wird.
Kontrolle: Die Entscheidung ist einer Rolle zugeordnet und dokumentiert.

Was dadurch möglich wird

- Einen falschen Wert einem Schritt, einem Prompt und einer Modellversion zuordnen.
- Einzelne Schritte durch Code, ein anderes Verfahren oder ein kleineres Modell ersetzen.
- Die Zahl der Modellaufrufe je Vorgang kennen, statt sie zu schätzen.
- Jede Änderung gegen eine Prüfmenge testen, bevor sie in den Betrieb geht.

DATALYXT

EINORDNUNG

Wann ein einzelner Aufruf genügt

Die Aufteilung kostet Aufwand. In diesen vier Fällen lohnt sie sich nicht.

- **Die Aufgabe ist tatsächlich ein Schritt**

  Zusammenfassen, umformulieren, übersetzen, einordnen. Dahinter steckt keine Kette.

- **Das Ergebnis liest ein Mensch und sonst niemand**

  Es wird nicht weiterverarbeitet, gespeichert oder in ein Folgedokument übernommen.

- **Ein Fehler kostet nichts**

  Ein unpassender Vorschlag wird verworfen, und damit ist die Sache erledigt.

- **Es geht um einen Versuch**

  Erst prüfen, ob die Aufgabe überhaupt lösbar ist. Danach wird aufgeteilt, nicht vorher.

Was ein Prompt nicht leisten kann

- **Er erzwingt keine Reihenfolge**

  Er legt sie nahe. Ob sie eingehalten wird, entscheidet sich in jedem Lauf neu.

- **Er ersetzt keine Rechnung**

  Eine Formel im Prompt wird nachgeahmt, nicht ausgeführt.

- **Seine Wirkung lässt sich nicht versionieren**

  Dieselbe Änderung hilft in einem Fall und schadet im nächsten. Sichtbar wird das erst im Betrieb.

- **Er lässt sich nicht testen wie Code**

  Es gibt keine Zusicherung, sondern eine Trefferquote über eine Menge von Fällen.

Die Frage ist nicht, ob ein Sprachmodell die Aufgabe lösen kann.
Die Frage ist, woran Sie merken, dass es sie in diesem Fall nicht gelöst hat.

Ein einzelner Aufruf gibt darauf keine Antwort. Eine Kette aus geprüften Schritten schon.